

PATTERN RECOGNITION USING HIGHER-ORDER LOCAL AUTOCORRELATION COEFFICIENTS

Vlad Popovici and Jean-Philippe Thiran
Signal Processing Institute (ITS-LTS1)
Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland
Phone: +41 21 693 5646,4623
Fax: +41 21 693 7600
E-mail: Vlad.Popovici,JP.Thiran@epfl.ch
Web: <http://ltswww.epfl.ch>

Abstract. The autocorrelations have been previously used as features for $1D$ or $2D$ signal classification in a wide range of applications, like texture classification, face detection and recognition, EEG signal classification, and so on. However, in almost all the cases, the high computational costs have hampered the extension to higher orders (more than the second order). In this paper we present a method which avoids the computation of the autocorrelation coefficients and which can be applied to a large set of tools commonly used in statistical pattern recognition. We will discuss different scenarios of using the autocorrelations and we will show that the order of autocorrelations is no longer an obstacle.

INTRODUCTION

Usually, in the framework of statistical pattern recognition, one pattern can be viewed as a function of time and/or spatial coordinates. Moreover, in most cases the class membership does not change as the pattern is translated or scaled. In such situations we would like to design a classifier that is invariant to a given class of affine transformations. One approach consists in transforming each pattern through a function which would induce this invariancy: if we consider each pattern as a point in a vector space, we wish to map all points corresponding to translated (and/or scaled) versions of one pattern in a single point. In addition, patterns which differ in other ways should map into distinct points, and in some sense, patterns which are similar should map into points that are close together.

As it will be shown, the autocorrelation functions possess the uniqueness property for even orders ([14]) and they are translation invariant. The autocorrelations

This work is supported by the BANCA project of the IST European program with the financial support of the Swiss OFES.

have been used in a wide range of applications: character recognition ([14]), geo-spatial data mining ([4]), affine-invariant texture classification ([6]), time series classification ([10]) and face detection and recognition ([15], [9], [7]). However, in most cases, the applicability of the autocorrelations has been limited to first or second order, due to high computational costs. An interesting approach is presented in ([11]) where the authors use the autocorrelations up to the third order and obtain a scale invariant classification by integrating over different scales. They succeed to reduce the number of computations by using the shift-invariant property of the autocorrelations and by a priori determining the lags for which the autocorrelations are not equivalent. The same approach is taken in ([12], [9]), but it has the disadvantage of not being easily extendable for higher orders or larger local domains.

Our interest is in finding an approach which scales well with the increase of the autocorrelation order, allowing us to generalize the use of autocorrelations. We will present the properties of the autocorrelation functions and will explore different approaches for autocorrelation-based pattern recognition. The paper is organised as follows: Section 2 discusses the properties of the autocorrelation function and sets the background for the following sections, where the Principal Component Analysis (Section 3) and kernel-based (Section 4) methods are applied to autocorrelation feature vectors.

GENERALIZED AUTOCORRELATION FUNCTIONS

Definition and Properties

Let $\psi : D \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$ be a real-valued function. The n -order autocorrelation function associated with function ψ is defined as (see Figure 2.1 for an example):

$$r_{\psi}^{(n)}(\tau_1, \dots, \tau_n) \triangleq \int \psi(t) \prod_{k=1}^n \psi(t + \tau_k) dt \quad (1)$$

It is easy to see that $r_{\psi}^{(n)}$ is shift-invariant, in the sense that $\psi(t)$ and $\psi(t + \tau)$ have the same n -th order autocorrelation.

On the other hand, for two functions ψ_1 and ψ_2 it can be proven ([14]) that the second order (and higher even order) autocorrelation functions are equal only if $\psi_1(t) = \psi_2(t + \tau)$, meaning that the two patterns have the same representation in autocorrelation space if ψ_2 is a shifted version of ψ_1 .

This also means that ψ_1 can be recovered from $r_{\psi_1}^{(2k)}$ except for an unknown translation τ . Generally, in the case of pattern recognition, this is a valuable property.

Considering the set of admissible values for τ_k being discrete and having m_k distinct values, it follows that the space of n -th order autocorrelations has $\prod_{k=1}^n m_k$ dimensions, making the explicit computation of autocorrelations prohibitively expensive.

Inner products of autocorrelations

The inner product of two autocorrelation functions is given by (see also Appendix A):

$$\langle \mathbf{r}_1, \mathbf{r}_2 \rangle = \int \left\{ \int \psi_1(v) \psi_2(v+s) dv \right\}^{n+1} ds \quad (2)$$

In the following, we will investigate the properties of those vectors, using the discrete version of (2):

$$\langle \mathbf{r}_1, \mathbf{r}_2 \rangle = \sum_{\tau} \left\{ \sum_t \psi_1(t) \psi_2(t+\tau) \right\}^{n+1} \quad (3)$$

For any ψ , the values of $r_{\psi}^{(n)}(\tau_1, \dots, \tau_n)$ can be ordered sequentially (for example, by letting the variables τ_k run faster than τ_i over the set of admissible values, for any $k > i$), obtaining a (column) vector $\mathbf{r}_{\psi}^{(n)}$. To simplify the notation, in the following we will denote by \mathbf{r}_k the n -th order autocorrelation vector corresponding to the function ψ_k .

Let now $\{\mathbf{r}_1, \dots, \mathbf{r}_m\}$ be a set of linearly independent autocorrelation vectors (not necessarily orthogonal), let $R = [\mathbf{r}_1 | \dots | \mathbf{r}_m]$ be the transformation matrix having these vectors as its columns, and let \mathbf{r} be a new vector to be projected on the space spanned by $\{\mathbf{r}_k\}_{k=1}^m$. The vector \mathbf{r} can be decomposed into two components:

$$\mathbf{r} = \mathbf{r}_W + \mathbf{r}_{\perp W} \quad (4)$$

where $\mathbf{r}_W \in W = \text{Span}(\{\mathbf{r}_k\})$ and $\mathbf{r}_{\perp W}$ is orthogonal on W . Then, the orthogonal projection \mathbf{r}_W onto W is given by (see Appendix B for a derivation of this result):

$$\mathbf{r}_W = R(R'R)^{-1}R'\mathbf{r} \quad (5)$$

Note that all products $R'\mathbf{r}$ and $R'R$ imply only computations of inner products between autocorrelation vectors which can be computed by means of (2)-(3), avoiding the explicit computation of autocorrelations. This method of avoiding the explicit computation of autocorrelation vectors is similar to the *kernel trick*, used, for example, in the context of Support Vector Machines ([19]).

Extended feature vectors

Combining autocorrelation of different orders in order to obtain a more descriptive feature vector can be done as follows. Let $I = \{i_1, \dots, i_m\}$ be a set of indices and let $\mathbf{r}_{\psi}^{(I)}$ be the vector obtained by concatenating the autocorrelations $\mathbf{r}_{\psi}^{(k)}$, where $k = i_1, \dots, i_m$, then it is obvious that

$$\langle \mathbf{r}_{\psi_1}^{(I)}, \mathbf{r}_{\psi_2}^{(I)} \rangle = \sum_{k=i_1, \dots, i_m} \langle \mathbf{r}_{\psi_1}^{(k)}, \mathbf{r}_{\psi_2}^{(k)} \rangle \quad (6)$$

meaning that computing the inner product of two compound feature vectors can be done by simply summing the inner products of the components.

Another consequence of this observation is the fact that the autocorrelations may be computed over any topology of the local neighborhood: one may consider a partition of the domain D and then use the local autocorrelation coefficients as discriminant features, still one can use (6) to compute the inner products. All the methods described bellow use the simple autocorrelations vectors, but can be applied directly to extended autocorrelation vectors as well.

APPLYING PCA TO AUTOCORRELATION FEATURE VECTORS

Principal Component Analysis (PCA) is a technique for extracting the structure from a high-dimensional data set. PCA can be viewed as an orthogonal transformation of the coordinate system in which the data is described. This transformation is performed in the hope that a small number of principal directions will suffice to well-approximate the data. There are different methods to perform PCA, the most common requiring the diagonalization of the covariance matrix, or, equivalently, to solve the eigenproblem

$$C\lambda_i = \lambda_i \mathbf{v}_i \quad (7)$$

where C is the covariance matrix and λ_i are the eigenvalues corresponding to the eigenvectors \mathbf{v}_i . Naturally, we are interested only in the non-trivial solution of (7).

Let $\{\mathbf{r}_k\}$ be a set of mean-centered autocorrelation vectors. The equivalent problem of (7) is

$$RR'\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad (8)$$

where the elements of the matrix RR' are formed by outer products $\mathbf{r}_i \mathbf{r}_j'$. The rank of RR' cannot exceed m , the number of data/autocorrelation vectors, even if its dimensionality is usually much bigger than $m \times m$. We can solve the problem (8) indirectly, by first solving the smaller problem

$$R'R\mathbf{w}_i = \lambda_i \mathbf{w}_i. \quad (9)$$

By left-multiplying with R we obtain

$$(RR')(\mathbf{Rw}_i) = \lambda_i (\mathbf{Rw}_i) \quad (10)$$

If $\lambda_i \neq 0$ (the only case we are interested in), then

$$\mathbf{v}_i = \frac{R\mathbf{w}_i}{\sqrt{\lambda_i}} \quad (11)$$

is the solution of (8) when \mathbf{w}_i is the solution of (9). Since the ranks of RR' and $R'R$ are equal, there are no eigenvectors missed or added by this indirect method.

Then, the projections of the vectors $\{\mathbf{r}_k\}_k$ on the principal directions will be given by

$$\mathbf{a}_i = R' \mathbf{v}_i = \frac{R' R \mathbf{w}_i}{\sqrt{\lambda_i}} \quad (12)$$

Generally, \mathbf{v}_i are not valid n -th order autocorrelations so the projection of a vector \mathbf{r} on \mathbf{v}_i cannot be computed directly as a simple inner product. Instead we have to use

$$\mathbf{r}' \mathbf{v}_i = \frac{\mathbf{r}' R \mathbf{w}_i}{\sqrt{\lambda_i}} \quad (13)$$

All of the above development has been done supposing that the vectors \mathbf{r}_k are centered around their mean. We will remove now this restriction and we will prove that the centering in the autocorrelation space can be carried out indirectly, without computing the autocorrelations. In (8-13) we have to replace the matrix R with R_* where $R_* = [\mathbf{r}_1 - \bar{\mathbf{r}} | \dots | \mathbf{r}_m - \bar{\mathbf{r}}]$, with $\bar{\mathbf{r}}$ being the mean autocorrelation vector. Computing the product $R'_* R_*$ reduces to compute the inner products $\langle \mathbf{r}_i - \bar{\mathbf{r}}, \mathbf{r}_j - \bar{\mathbf{r}} \rangle$ for all $i, j = 1, \dots, m$:

$$\begin{aligned} \langle \mathbf{r}_i - \bar{\mathbf{r}}, \mathbf{r}_j - \bar{\mathbf{r}} \rangle &= \\ &= \langle \mathbf{r}_i, \mathbf{r}_j \rangle - \frac{1}{m} \sum_{k=1}^m \langle \mathbf{r}_i, \mathbf{r}_k \rangle \\ &\quad - \frac{1}{m} \sum_{k=1}^m \langle \mathbf{r}_j, \mathbf{r}_k \rangle + \frac{1}{m^2} \sum_{k,l=1}^m \langle \mathbf{r}_k, \mathbf{r}_l \rangle \end{aligned} \quad (14)$$

which translates into

$$\begin{aligned} R'_* R_* &= R' R - \frac{1}{m} \mathbf{1}_{mm} (R' R) - \frac{1}{m} (R' R) \mathbf{1}_{mm} \\ &\quad + \frac{1}{m^2} \mathbf{1}_{mm} (R' R) \mathbf{1}_{mm} \end{aligned} \quad (15)$$

where $\mathbf{1}_{mm}$ is an $m \times m$ matrix of ones.

Finally, we have to compute the projection of $\mathbf{r}_* = \mathbf{r}_0 - \bar{\mathbf{r}}$ on the principal axis, where \mathbf{r}_0 is a new autocorrelation vector which has to be projected on the principal directions. Similar to (15), we have:

$$\begin{aligned} \mathbf{r}'_* R_* &= \mathbf{r}'_0 R - \frac{1}{m} \mathbf{1}_{1m} (R' R) - \frac{1}{m} (\mathbf{r}'_0 R) \mathbf{1}_{mm} \\ &\quad + \frac{1}{m^2} \mathbf{1}_{1m} (R' R) \mathbf{1}_{mm} \end{aligned} \quad (16)$$

and from (13) we have the projection on the i -th principal direction:

$$\mathbf{r}'_* \mathbf{v}_{*i} = \frac{\mathbf{r}'_* R_* \mathbf{w}_{*i}}{\sqrt{\lambda_{*i}}} \quad (17)$$

where $\mathbf{v}_{*i}, \mathbf{w}_{*i}$ and λ_{*i} are obtained by considering equations (9) and (11) with R_* replaced for R .

HIGHER-ORDER AUTOCORRELATIONS IN THE CONTEXT OF KERNEL-BASED METHODS

A standard technique of transforming a linear classifier into a nonlinear one, consists in projecting the initial space into a feature space, through a non-linear mapping $\Phi: X \rightarrow K$. Now, being given a fixed mapping $\Phi: X \rightarrow K$, we define the *kernel* function as the inner product function $k: X \times X \rightarrow \mathbb{R}$, i.e., for all $\mathbf{x}_1, \mathbf{x}_2 \in X$:

$$k(\mathbf{x}_1, \mathbf{x}_2) \triangleq \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle \quad (18)$$

For a set of m vectors $\mathbf{x}_i \in X^m$, the Gram matrix

$$G_{ij} \triangleq \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j) \quad (19)$$

is called *kernel matrix*.

Usually we are not given the function $\Phi(\cdot)$, but the kernel function $k(\cdot, \cdot)$. Some of the kernels used in practice are

Polynomial kernel

$$k_P(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^p$$

Sigmoidal kernel

$$k_S(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \delta)$$

Radial Basis Function kernel

$$k_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) = \exp(-\gamma(\langle \mathbf{x}_i, \mathbf{x}_i \rangle + \langle \mathbf{x}_j, \mathbf{x}_j \rangle - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle))$$

It follows that for all kernel functions that can be expressed in terms of inner products of data, we can use the technique developed above to carry out the computations of the kernel matrix. Then, performing a kernel PCA ([18]) or training a Support Vector Machine ([19],[5]) is an immediate task ([15],[16]).

EXPERIMENTS

In order to assess the validity of the method presented here, we carried out a number of tests on the dataset *Waveform* from the UCI database ([1]). The set consists of 5000 samples of 1D signals, distributed equally in 3 classes. The goal of the experiment was to study the influence of different parameters in a binary classification task: discriminate between the first class (called *A*) and the other two (*B* and *C*).

The discrimination function was based on the *distance from feature space (DFFS)*: a sample is classified as belonging to class A if its *DFFS* to the feature space of class A is less than a threshold.

In all experiments, 500 vectors from class A (randomly chosen) have been used to perform PCA in autocorrelation space, and to determine the threshold. Other 500 vectors from class A and 500 from classes B and C have been used to test the classification. Table 1 presents some results obtained by autocorrelations features in comparison with some other methods (notation $\text{ACorr}(n,d)$ is used to designate an autocorrelation function of order n having d distinct values for each of the variables τ_i (lags)).

Table 1: Classification rates (%)

Method	Rate
Optimal Bayes classifier ([1])	86.0%
Nearest Neighbor ([1])	78.0%
CART decision tree ([1])	72.0%
$\text{ACorr}(2,5)$	80.5%
$\text{ACorr}(2,7)$	79.0%
$\text{ACorr}(3,5)$	81.6%
$\text{ACorr}(3,7)$	78.0%
$\text{ACorr}(4,7)$	81.5%
$\text{ACorr}(4,7)$	80.0%

APPENDICES

Appendix A Inner product of two autocorrelation vectors

Let ψ_1 and ψ_2 be two functions defined on the same domain $D \subset \mathbb{R}^m$. Then the inner product of the autocorrelation functions can be derived as follows ([14], [15]):

$$\begin{aligned}
\langle \mathbf{r}_1, \mathbf{r}_2 \rangle &= \\
&= \int \dots \int \mathbf{r}_1(\tau_1, \dots, \tau_n) \cdot \mathbf{r}_2(\tau_1, \dots, \tau_n) d\tau_1 \dots d\tau_n \\
&= \int \dots \int \left\{ \int \psi_1(t) \psi_1(t + \tau_1) \dots \psi_1(t + \tau_n) du \right\} \\
&\quad \cdot \left\{ \int \psi_2(u) \psi_2(u + \tau_1) \dots \psi_2(u + \tau_n) du \right\} d\tau_1 \dots d\tau_n \\
&= \int \int \psi_1(t) \psi_2(u) \left\{ \int \psi_1(t + \tau) \psi_2(u + \tau) d\tau \right\}^n du dt \\
&= \int \int \psi_1(t) \psi_2(s + t) \left\{ \int \psi_1(v) \psi_2(v + s) dv \right\}^n ds dt \\
&= \int \left\{ \int \psi_1(v) \psi_2(v + s) dv \right\}^{n+1} ds
\end{aligned}$$

Appendix B Orthogonal projection of autocorrelation vectors

Using the notations from section 2.2, any vector \mathbf{r} can be written as a sum of two components: $\mathbf{r} = \mathbf{r}_W + \mathbf{r}_{\perp W}$. Since $\mathbf{r}_W \in W$, it is a linear combination of vectors $\mathbf{r}_1, \dots, \mathbf{r}_m$, so there exists a vector $\mathbf{c} \in \mathbb{R}^m$ such as

$$\mathbf{r}_W = R\mathbf{c}$$

From the fact that $\mathbf{r}_{\perp W}$ is orthogonal on W we have

$$R'\mathbf{r} = R'(\mathbf{r}_W + \mathbf{r}_{\perp W}) = R'\mathbf{r}_W = R'R\mathbf{c}$$

It follows that

$$\begin{aligned}\mathbf{c} &= (R'R)^{-1}R'\mathbf{r} \\ \mathbf{r}_W &= R(R'R)^{-1}R'\mathbf{r}\end{aligned}$$

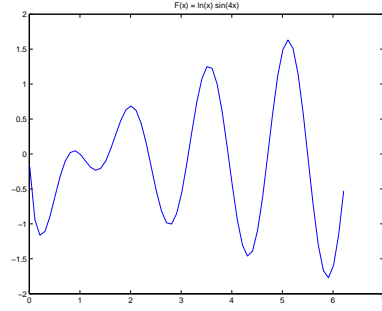
Thus we have the orthogonal projection of \mathbf{r} on W . Further, we can obtain the modulus of the projection and the distance from the space W by

$$\begin{aligned}\|\mathbf{r}_W\|^2 &= \langle \mathbf{r}_W, \mathbf{r}_W \rangle = (R'\mathbf{r})'(R'R)^{-1}(R'\mathbf{r}) \\ \|\mathbf{r}_{\perp W}\|^2 &= \|\mathbf{r}\|^2 - \|\mathbf{r}_W\|^2 = \\ &\langle \mathbf{r}, \mathbf{r} \rangle - (R'\mathbf{r})'(R'R)^{-1}(R'\mathbf{r})\end{aligned}$$

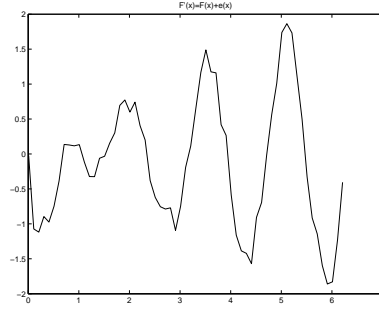
REFERENCES

- [1] C. Blake and C. Merz, "UCI Repository of machine learning databases," .
- [2] T. Breuel, "Higher-Order Statistics in Visual Object Recognition," Memo 93-02, **IDIAP**, June 1993.
- [3] M. Á. Carreira-Perpiñán, "A Review of Dimension Reduction Techniques," Technical Report CS-96-09, **Dept. of Computer Science, University of Sheffield**, January 1997.
- [4] S. Chawla, S. Shekhar, W. Wu and U. Ozesmi, "Modeling spatial dependencies for mining geospatial data: An introduction," 2000.
- [5] N. Cristianini and J. Shawe-Taylor, **An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods**, Cambridge University Press, 2000.
- [6] D. Chetverikov and Z. Foldvari, "Affine-Invariant Texture Classification Using Regularity Features," in M. Pietikainen (ed.), **Texture Analysis in Machine Vision**, World Scientific, 2000, Series in Machine Perception and Artificial Intelligence.
- [7] F. Goudail, E. Lange, T. Iwamoto, K. Kyuma and N. Otsu, "Face Recognition System Using Local Autocorrelations and Multiscale Integration," **IEEE Trans. on Pattern Analysis and Machine Intelligence**, vol. 18, no. 10, pp. 1024 – 1028, October 1996.
- [8] R. Herbrich, **Learning Kernel Classifiers: Theory and Algorithms**, Adaptive Computation and Machine Learning, The MIT Press, 2000.
- [9] K. Hotta, T. Kurita and T. Mishima, "Scale Invariant Face Detection Method using Higher-Order Local Autocorrelation Features extracted from Log-Polar Image," in **Automatic Face and Gesture Recognition. Proceedings. Third IEEE International Conference on**, IEEE, 1998, pp. 70–75.

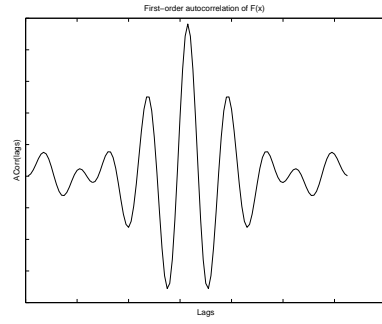
- [10] E. J. Keogh and M. J. Pazzani, "An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback," in **Knowledge Discovery and Data Mining**, 1998, pp. 239–243.
- [11] M. Kreutz, B. Volpel and H. Janssen, "Scale-Invariant Image Recognition Based on Higher Order Autocorrelation Features," **Pattern Recognition**, vol. 29, no. 1, 1996.
- [12] T. Kurita, K. Hotta and T. Mishima, "Scale and Rotation Invariant Recognition Method Using Higher-Order Local Autocorrelation Features of Log-Polar Image," in **Third Asian Conference on Computer Vision**, 1998, pp. 89–96.
- [13] T. Kurita, N. Otsu and T. Sato, "A Face Recognition Method Using Higher Order Local Autocorrelation and Multivariate Analysis," in **11th IAPR International Conference on Pattern Recognition**, The Hague, 1992, pp. 213 – 216.
- [14] J. A. McLaughlin and J. Raviv, "Nth-order autocorrelations in pattern recognition," **Information and Control**, vol. 12, pp. 121 – 142, 1968.
- [15] V. Popovici and J. P. Thiran, "Higher Order Autocorrelations for Pattern Classification," in **Proceedings of the International Conference on Image Processing (ICIP)**, IEEE, 2001.
- [16] V. Popovici and J.-P. Thiran, "PCA in Autocorrelation Space," in **International Conference on Pattern Recognition**, 2002, *to appear*.
- [17] S. Roweis, "EM Algorithms for PCA and SPCA," in M. I. Jordan, M. J. Kearns and S. A. Solla (eds.), **Advances in Neural Information Processing Systems**, The MIT Press, 1998, vol. 10.
- [18] B. Schölkopf, S. Mika, A. Smola, G. Rätsch and K.-R. Müller, "Kernel PCA Pattern Reconstruction *via* Approximate Pre-Images," in L. Niklasson, M. Bodén and T. Ziemke (eds.), **Proceedings of the 8th International Conference on Artificial Neural Networks**, Springer Verlag, 1998, pp. 147–152.
- [19] V. Vapnik, **The Nature of Statistical Learning Theory**, Springer Verlag, 1995.



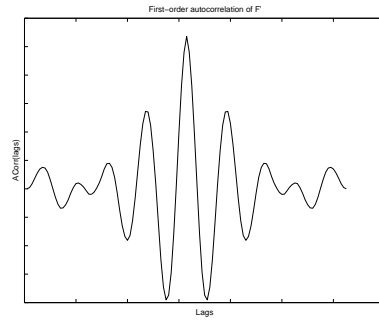
(a) $F(x) = \ln x \sin(4x)$



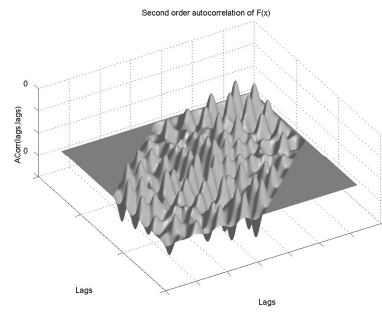
(b) $F'(x) = F(x) + \epsilon(x)$



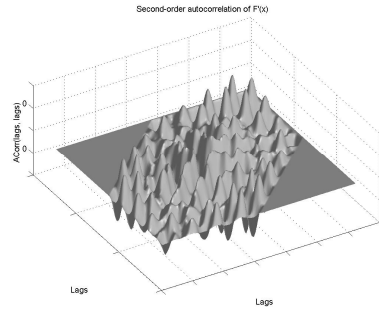
(c) First order autocorrelation of $F(x)$



(d) First order autocorrelation of $F'(x)$



(e) Second order autocorrelation of $F(x)$



(f) Second order autocorrelation of $F'(x)$

Figure 1: First and second order autocorrelations of a function $F(x)$ and of its noisy version $F'(x)$. Note how the autocorrelations are less affected by the noise, than the original representation of the function.